| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultTreeSelectionModel.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/tree/DefaultTreeModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/tree/ExpandVetoException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/tree/DefaultTreeSelectionModel.html)    [**NO FRAMES**](http://docs.google.com/DefaultTreeSelectionModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#3j2qqm3) | [METHOD](#4i7ojhp) |

## **javax.swing.tree**

Class DefaultTreeSelectionModel

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.swing.tree.DefaultTreeSelectionModel**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) **Direct Known Subclasses:** [JTree.EmptySelectionModel](http://docs.google.com/javax/swing/JTree.EmptySelectionModel.html)

public class **DefaultTreeSelectionModel**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

Default implementation of TreeSelectionModel. Listeners are notified whenever the paths in the selection change, not the rows. In order to be able to track row changes you may wish to become a listener for expansion events on the tree and test for changes from there.

resetRowSelection is called from any of the methods that update the selected paths. If you subclass any of these methods to filter what is allowed to be selected, be sure and message resetRowSelection if you do not message super.

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeansTM has been added to the java.beans package. Please see [XMLEncoder](http://docs.google.com/java/beans/XMLEncoder.html).

**See Also:**[JTree](http://docs.google.com/javax/swing/JTree.html)

| **Field Summary** | |
| --- | --- |
| protected  [SwingPropertyChangeSupport](http://docs.google.com/javax/swing/event/SwingPropertyChangeSupport.html) | [**changeSupport**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#changeSupport)            Used to messaged registered listeners. |
| protected  int | [**leadIndex**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#leadIndex)            Index of the lead path in selection. |
| protected  [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) | [**leadPath**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#leadPath)            Last path that was added. |
| protected  int | [**leadRow**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#leadRow)            Lead row. |
| protected  [EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html) | [**listenerList**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#listenerList)            Event listener list. |
| protected  [DefaultListSelectionModel](http://docs.google.com/javax/swing/DefaultListSelectionModel.html) | [**listSelectionModel**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#listSelectionModel)            Handles maintaining the list selection model. |
| protected  [RowMapper](http://docs.google.com/javax/swing/tree/RowMapper.html) | [**rowMapper**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#rowMapper)            Provides a row for a given path. |
| protected  [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] | [**selection**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#selection)            Paths that are currently selected. |
| static [String](http://docs.google.com/java/lang/String.html) | [**SELECTION\_MODE\_PROPERTY**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#SELECTION_MODE_PROPERTY)            Property name for selectionMode. |
| protected  int | [**selectionMode**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#selectionMode)            Mode for the selection, will be either SINGLE\_TREE\_SELECTION, CONTIGUOUS\_TREE\_SELECTION or DISCONTIGUOUS\_TREE\_SELECTION. |

| **Fields inherited from interface javax.swing.tree.**[**TreeSelectionModel**](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) |
| --- |
| [CONTIGUOUS\_TREE\_SELECTION](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#CONTIGUOUS_TREE_SELECTION), [DISCONTIGUOUS\_TREE\_SELECTION](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#DISCONTIGUOUS_TREE_SELECTION), [SINGLE\_TREE\_SELECTION](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#SINGLE_TREE_SELECTION) |

| **Constructor Summary** | |
| --- | --- |
| [**DefaultTreeSelectionModel**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#DefaultTreeSelectionModel())()            Creates a new instance of DefaultTreeSelectionModel that is empty, with a selection mode of DISCONTIGUOUS\_TREE\_SELECTION. |

| **Method Summary** | |
| --- | --- |
| void | [**addPropertyChangeListener**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#addPropertyChangeListener(java.beans.PropertyChangeListener))([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)            Adds a PropertyChangeListener to the listener list. |
| void | [**addSelectionPath**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#addSelectionPath(javax.swing.tree.TreePath))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)            Adds path to the current selection. |
| void | [**addSelectionPaths**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#addSelectionPaths(javax.swing.tree.TreePath%5B%5D))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] paths)            Adds paths to the current selection. |
| void | [**addTreeSelectionListener**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#addTreeSelectionListener(javax.swing.event.TreeSelectionListener))([TreeSelectionListener](http://docs.google.com/javax/swing/event/TreeSelectionListener.html) x)            Adds x to the list of listeners that are notified each time the set of selected TreePaths changes. |
| protected  boolean | [**arePathsContiguous**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#arePathsContiguous(javax.swing.tree.TreePath%5B%5D))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] paths)            Returns true if the paths are contiguous, or this object has no RowMapper. |
| protected  boolean | [**canPathsBeAdded**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#canPathsBeAdded(javax.swing.tree.TreePath%5B%5D))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] paths)            Used to test if a particular set of TreePaths can be added. |
| protected  boolean | [**canPathsBeRemoved**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#canPathsBeRemoved(javax.swing.tree.TreePath%5B%5D))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] paths)            Returns true if the paths can be removed without breaking the continuity of the model. |
| void | [**clearSelection**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#clearSelection())()            Empties the current selection. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#clone())()            Returns a clone of this object with the same selection. |
| protected  void | [**fireValueChanged**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#fireValueChanged(javax.swing.event.TreeSelectionEvent))([TreeSelectionEvent](http://docs.google.com/javax/swing/event/TreeSelectionEvent.html) e)            Notifies all listeners that are registered for tree selection events on this object. |
| [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) | [**getLeadSelectionPath**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getLeadSelectionPath())()            Returns the last path that was added. |
| int | [**getLeadSelectionRow**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getLeadSelectionRow())()            Returns the lead selection index. |
| | <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)>  T[] | | --- | | [**getListeners**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getListeners(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)            Returns an array of all the objects currently registered as *Foo*Listeners upon this model. |
| int | [**getMaxSelectionRow**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getMaxSelectionRow())()            Returns the largest value obtained from the RowMapper for the current set of selected TreePaths. |
| int | [**getMinSelectionRow**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getMinSelectionRow())()            Returns the smallest value obtained from the RowMapper for the current set of selected TreePaths. |
| [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html)[] | [**getPropertyChangeListeners**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getPropertyChangeListeners())()            Returns an array of all the property change listeners registered on this DefaultTreeSelectionModel. |
| [RowMapper](http://docs.google.com/javax/swing/tree/RowMapper.html) | [**getRowMapper**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getRowMapper())()            Returns the RowMapper instance that is able to map a TreePath to a row. |
| int | [**getSelectionCount**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getSelectionCount())()            Returns the number of paths that are selected. |
| int | [**getSelectionMode**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getSelectionMode())()            Returns the selection mode, one of SINGLE\_TREE\_SELECTION, DISCONTIGUOUS\_TREE\_SELECTION or CONTIGUOUS\_TREE\_SELECTION. |
| [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) | [**getSelectionPath**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getSelectionPath())()            Returns the first path in the selection. |
| [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] | [**getSelectionPaths**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getSelectionPaths())()            Returns the paths in the selection. |
| int[] | [**getSelectionRows**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getSelectionRows())()            Returns all of the currently selected rows. |
| [TreeSelectionListener](http://docs.google.com/javax/swing/event/TreeSelectionListener.html)[] | [**getTreeSelectionListeners**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getTreeSelectionListeners())()            Returns an array of all the tree selection listeners registered on this model. |
| protected  void | [**insureRowContinuity**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#insureRowContinuity())()            Makes sure the currently selected TreePaths are valid for the current selection mode. |
| protected  void | [**insureUniqueness**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#insureUniqueness())()            This method is obsolete and its implementation is now a noop. |
| boolean | [**isPathSelected**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#isPathSelected(javax.swing.tree.TreePath))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)            Returns true if the path, path, is in the current selection. |
| boolean | [**isRowSelected**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#isRowSelected(int))(int row)            Returns true if the row identified by row is selected. |
| boolean | [**isSelectionEmpty**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#isSelectionEmpty())()            Returns true if the selection is currently empty. |
| protected  void | [**notifyPathChange**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#notifyPathChange(java.util.Vector,%20javax.swing.tree.TreePath))([Vector](http://docs.google.com/java/util/Vector.html)<javax.swing.tree.PathPlaceHolder> changedPaths, [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) oldLeadSelection)            Notifies listeners of a change in path. |
| void | [**removePropertyChangeListener**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#removePropertyChangeListener(java.beans.PropertyChangeListener))([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)            Removes a PropertyChangeListener from the listener list. |
| void | [**removeSelectionPath**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#removeSelectionPath(javax.swing.tree.TreePath))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)            Removes path from the selection. |
| void | [**removeSelectionPaths**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#removeSelectionPaths(javax.swing.tree.TreePath%5B%5D))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] paths)            Removes paths from the selection. |
| void | [**removeTreeSelectionListener**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#removeTreeSelectionListener(javax.swing.event.TreeSelectionListener))([TreeSelectionListener](http://docs.google.com/javax/swing/event/TreeSelectionListener.html) x)            Removes x from the list of listeners that are notified each time the set of selected TreePaths changes. |
| void | [**resetRowSelection**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#resetRowSelection())()            Updates this object's mapping from TreePath to rows. |
| void | [**setRowMapper**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#setRowMapper(javax.swing.tree.RowMapper))([RowMapper](http://docs.google.com/javax/swing/tree/RowMapper.html) newMapper)            Sets the RowMapper instance. |
| void | [**setSelectionMode**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#setSelectionMode(int))(int mode)            Sets the selection model, which must be one of SINGLE\_TREE\_SELECTION, CONTIGUOUS\_TREE\_SELECTION or DISCONTIGUOUS\_TREE\_SELECTION. |
| void | [**setSelectionPath**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#setSelectionPath(javax.swing.tree.TreePath))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)            Sets the selection to path. |
| void | [**setSelectionPaths**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#setSelectionPaths(javax.swing.tree.TreePath%5B%5D))([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] pPaths)            Sets the selection to the paths in paths. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#toString())()            Returns a string that displays and identifies this object's properties. |
| protected  void | [**updateLeadIndex**](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#updateLeadIndex())()            Updates the leadIndex instance variable. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### SELECTION\_MODE\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **SELECTION\_MODE\_PROPERTY**

Property name for selectionMode.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.tree.DefaultTreeSelectionModel.SELECTION_MODE_PROPERTY)

### changeSupport

protected [SwingPropertyChangeSupport](http://docs.google.com/javax/swing/event/SwingPropertyChangeSupport.html) **changeSupport**

Used to messaged registered listeners.

### selection

protected [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] **selection**

Paths that are currently selected. Will be null if nothing is currently selected.

### listenerList

protected [EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html) **listenerList**

Event listener list.

### rowMapper

protected transient [RowMapper](http://docs.google.com/javax/swing/tree/RowMapper.html) **rowMapper**

Provides a row for a given path.

### listSelectionModel

protected [DefaultListSelectionModel](http://docs.google.com/javax/swing/DefaultListSelectionModel.html) **listSelectionModel**

Handles maintaining the list selection model. The RowMapper is used to map from a TreePath to a row, and the value is then placed here.

### selectionMode

protected int **selectionMode**

Mode for the selection, will be either SINGLE\_TREE\_SELECTION, CONTIGUOUS\_TREE\_SELECTION or DISCONTIGUOUS\_TREE\_SELECTION.

### leadPath

protected [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) **leadPath**

Last path that was added.

### leadIndex

protected int **leadIndex**

Index of the lead path in selection.

### leadRow

protected int **leadRow**

Lead row.

| **Constructor Detail** |
| --- |

### DefaultTreeSelectionModel

public **DefaultTreeSelectionModel**()

Creates a new instance of DefaultTreeSelectionModel that is empty, with a selection mode of DISCONTIGUOUS\_TREE\_SELECTION.

| **Method Detail** |
| --- |

### setRowMapper

public void **setRowMapper**([RowMapper](http://docs.google.com/javax/swing/tree/RowMapper.html) newMapper)

Sets the RowMapper instance. This instance is used to determine the row for a particular TreePath.

**Specified by:**[setRowMapper](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#setRowMapper(javax.swing.tree.RowMapper)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### getRowMapper

public [RowMapper](http://docs.google.com/javax/swing/tree/RowMapper.html) **getRowMapper**()

Returns the RowMapper instance that is able to map a TreePath to a row.

**Specified by:**[getRowMapper](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#getRowMapper()) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### setSelectionMode

public void **setSelectionMode**(int mode)

Sets the selection model, which must be one of SINGLE\_TREE\_SELECTION, CONTIGUOUS\_TREE\_SELECTION or DISCONTIGUOUS\_TREE\_SELECTION. If mode is not one of the defined value, DISCONTIGUOUS\_TREE\_SELECTION is assumed.

This may change the selection if the current selection is not valid for the new mode. For example, if three TreePaths are selected when the mode is changed to SINGLE\_TREE\_SELECTION, only one TreePath will remain selected. It is up to the particular implementation to decide what TreePath remains selected.

Setting the mode to something other than the defined types will result in the mode becoming DISCONTIGUOUS\_TREE\_SELECTION.

**Specified by:**[setSelectionMode](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#setSelectionMode(int)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### getSelectionMode

public int **getSelectionMode**()

Returns the selection mode, one of SINGLE\_TREE\_SELECTION, DISCONTIGUOUS\_TREE\_SELECTION or CONTIGUOUS\_TREE\_SELECTION.

**Specified by:**[getSelectionMode](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#getSelectionMode()) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### setSelectionPath

public void **setSelectionPath**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)

Sets the selection to path. If this represents a change, then the TreeSelectionListeners are notified. If path is null, this has the same effect as invoking clearSelection.

**Specified by:**[setSelectionPath](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#setSelectionPath(javax.swing.tree.TreePath)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) **Parameters:**path - new path to select

### setSelectionPaths

public void **setSelectionPaths**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] pPaths)

Sets the selection to the paths in paths. If this represents a change the TreeSelectionListeners are notified. Potentially paths will be held by this object; in other words don't change any of the objects in the array once passed in.

If paths is null, this has the same effect as invoking clearSelection.

The lead path is set to the last path in pPaths.

If the selection mode is CONTIGUOUS\_TREE\_SELECTION, and adding the new paths would make the selection discontiguous, the selection is reset to the first TreePath in paths.

**Specified by:**[setSelectionPaths](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#setSelectionPaths(javax.swing.tree.TreePath%5B%5D)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) **Parameters:**pPaths - new selection

### addSelectionPath

public void **addSelectionPath**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)

Adds path to the current selection. If path is not currently in the selection the TreeSelectionListeners are notified. This has no effect if path is null.

**Specified by:**[addSelectionPath](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#addSelectionPath(javax.swing.tree.TreePath)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) **Parameters:**path - the new path to add to the current selection

### addSelectionPaths

public void **addSelectionPaths**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] paths)

Adds paths to the current selection. If any of the paths in paths are not currently in the selection the TreeSelectionListeners are notified. This has no effect if paths is null.

The lead path is set to the last element in paths.

If the selection mode is CONTIGUOUS\_TREE\_SELECTION, and adding the new paths would make the selection discontiguous. Then two things can result: if the TreePaths in paths are contiguous, then the selection becomes these TreePaths, otherwise the TreePaths aren't contiguous and the selection becomes the first TreePath in paths.

**Specified by:**[addSelectionPaths](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#addSelectionPaths(javax.swing.tree.TreePath%5B%5D)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) **Parameters:**paths - the new path to add to the current selection

### removeSelectionPath

public void **removeSelectionPath**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)

Removes path from the selection. If path is in the selection The TreeSelectionListeners are notified. This has no effect if path is null.

**Specified by:**[removeSelectionPath](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#removeSelectionPath(javax.swing.tree.TreePath)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) **Parameters:**path - the path to remove from the selection

### removeSelectionPaths

public void **removeSelectionPaths**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] paths)

Removes paths from the selection. If any of the paths in paths are in the selection the TreeSelectionListeners are notified. This has no effect if paths is null.

**Specified by:**[removeSelectionPaths](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#removeSelectionPaths(javax.swing.tree.TreePath%5B%5D)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) **Parameters:**paths - the paths to remove from the selection

### getSelectionPath

public [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) **getSelectionPath**()

Returns the first path in the selection. This is useful if there if only one item currently selected.

**Specified by:**[getSelectionPath](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#getSelectionPath()) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### getSelectionPaths

public [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] **getSelectionPaths**()

Returns the paths in the selection. This will return null (or an empty array) if nothing is currently selected.

**Specified by:**[getSelectionPaths](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#getSelectionPaths()) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### getSelectionCount

public int **getSelectionCount**()

Returns the number of paths that are selected.

**Specified by:**[getSelectionCount](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#getSelectionCount()) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### isPathSelected

public boolean **isPathSelected**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)

Returns true if the path, path, is in the current selection.

**Specified by:**[isPathSelected](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#isPathSelected(javax.swing.tree.TreePath)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### isSelectionEmpty

public boolean **isSelectionEmpty**()

Returns true if the selection is currently empty.

**Specified by:**[isSelectionEmpty](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#isSelectionEmpty()) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### clearSelection

public void **clearSelection**()

Empties the current selection. If this represents a change in the current selection, the selection listeners are notified.

**Specified by:**[clearSelection](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#clearSelection()) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### addTreeSelectionListener

public void **addTreeSelectionListener**([TreeSelectionListener](http://docs.google.com/javax/swing/event/TreeSelectionListener.html) x)

Adds x to the list of listeners that are notified each time the set of selected TreePaths changes.

**Specified by:**[addTreeSelectionListener](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#addTreeSelectionListener(javax.swing.event.TreeSelectionListener)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) **Parameters:**x - the new listener to be added

### removeTreeSelectionListener

public void **removeTreeSelectionListener**([TreeSelectionListener](http://docs.google.com/javax/swing/event/TreeSelectionListener.html) x)

Removes x from the list of listeners that are notified each time the set of selected TreePaths changes.

**Specified by:**[removeTreeSelectionListener](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#removeTreeSelectionListener(javax.swing.event.TreeSelectionListener)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) **Parameters:**x - the listener to remove

### getTreeSelectionListeners

public [TreeSelectionListener](http://docs.google.com/javax/swing/event/TreeSelectionListener.html)[] **getTreeSelectionListeners**()

Returns an array of all the tree selection listeners registered on this model.

**Returns:**all of this model's TreeSelectionListeners or an empty array if no tree selection listeners are currently registered**Since:** 1.4 **See Also:**[addTreeSelectionListener(javax.swing.event.TreeSelectionListener)](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#addTreeSelectionListener(javax.swing.event.TreeSelectionListener)), [removeTreeSelectionListener(javax.swing.event.TreeSelectionListener)](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#removeTreeSelectionListener(javax.swing.event.TreeSelectionListener))

### fireValueChanged

protected void **fireValueChanged**([TreeSelectionEvent](http://docs.google.com/javax/swing/event/TreeSelectionEvent.html) e)

Notifies all listeners that are registered for tree selection events on this object.

**See Also:**[addTreeSelectionListener(javax.swing.event.TreeSelectionListener)](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#addTreeSelectionListener(javax.swing.event.TreeSelectionListener)), [EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### getListeners

public <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)> T[] **getListeners**([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)

Returns an array of all the objects currently registered as *Foo*Listeners upon this model. *Foo*Listeners are registered using the add*Foo*Listener method.

You can specify the listenerType argument with a class literal, such as *Foo*Listener.class. For example, you can query a DefaultTreeSelectionModel m for its tree selection listeners with the following code:

TreeSelectionListener[] tsls = (TreeSelectionListener[])(m.getListeners(TreeSelectionListener.class));

If no such listeners exist, this method returns an empty array.

**Parameters:**listenerType - the type of listeners requested; this parameter should specify an interface that descends from java.util.EventListener **Returns:**an array of all objects registered as *Foo*Listeners on this component, or an empty array if no such listeners have been added **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if listenerType doesn't specify a class or interface that implements java.util.EventListener**Since:** 1.3 **See Also:**[getTreeSelectionListeners()](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getTreeSelectionListeners()), [getPropertyChangeListeners()](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#getPropertyChangeListeners())

### getSelectionRows

public int[] **getSelectionRows**()

Returns all of the currently selected rows. This will return null (or an empty array) if there are no selected TreePaths or a RowMapper has not been set. This may return an array of length less that than of the selected TreePaths if some of the rows are not visible (that is the RowMapper returned -1 for the row corresponding to the TreePath).

**Specified by:**[getSelectionRows](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#getSelectionRows()) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### getMinSelectionRow

public int **getMinSelectionRow**()

Returns the smallest value obtained from the RowMapper for the current set of selected TreePaths. If nothing is selected, or there is no RowMapper, this will return -1.

**Specified by:**[getMinSelectionRow](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#getMinSelectionRow()) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### getMaxSelectionRow

public int **getMaxSelectionRow**()

Returns the largest value obtained from the RowMapper for the current set of selected TreePaths. If nothing is selected, or there is no RowMapper, this will return -1.

**Specified by:**[getMaxSelectionRow](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#getMaxSelectionRow()) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### isRowSelected

public boolean **isRowSelected**(int row)

Returns true if the row identified by row is selected.

**Specified by:**[isRowSelected](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#isRowSelected(int)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### resetRowSelection

public void **resetRowSelection**()

Updates this object's mapping from TreePath to rows. This should be invoked when the mapping from TreePaths to integers has changed (for example, a node has been expanded).

You do not normally have to call this, JTree and its associated Listeners will invoke this for you. If you are implementing your own View class, then you will have to invoke this.

This will invoke insureRowContinuity to make sure the currently selected TreePaths are still valid based on the selection mode.

**Specified by:**[resetRowSelection](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#resetRowSelection()) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### getLeadSelectionRow

public int **getLeadSelectionRow**()

Returns the lead selection index. That is the last index that was added.

**Specified by:**[getLeadSelectionRow](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#getLeadSelectionRow()) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### getLeadSelectionPath

public [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) **getLeadSelectionPath**()

Returns the last path that was added. This may differ from the leadSelectionPath property maintained by the JTree.

**Specified by:**[getLeadSelectionPath](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#getLeadSelectionPath()) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html)

### addPropertyChangeListener

public void **addPropertyChangeListener**([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)

Adds a PropertyChangeListener to the listener list. The listener is registered for all properties.

A PropertyChangeEvent will get fired when the selection mode changes.

**Specified by:**[addPropertyChangeListener](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#addPropertyChangeListener(java.beans.PropertyChangeListener)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) **Parameters:**listener - the PropertyChangeListener to be added

### removePropertyChangeListener

public void **removePropertyChangeListener**([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)

Removes a PropertyChangeListener from the listener list. This removes a PropertyChangeListener that was registered for all properties.

**Specified by:**[removePropertyChangeListener](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html#removePropertyChangeListener(java.beans.PropertyChangeListener)) in interface [TreeSelectionModel](http://docs.google.com/javax/swing/tree/TreeSelectionModel.html) **Parameters:**listener - the PropertyChangeListener to be removed

### getPropertyChangeListeners

public [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html)[] **getPropertyChangeListeners**()

Returns an array of all the property change listeners registered on this DefaultTreeSelectionModel.

**Returns:**all of this model's PropertyChangeListeners or an empty array if no property change listeners are currently registered**Since:** 1.4 **See Also:**[addPropertyChangeListener(java.beans.PropertyChangeListener)](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [removePropertyChangeListener(java.beans.PropertyChangeListener)](http://docs.google.com/javax/swing/tree/DefaultTreeSelectionModel.html#removePropertyChangeListener(java.beans.PropertyChangeListener))

### insureRowContinuity

protected void **insureRowContinuity**()

Makes sure the currently selected TreePaths are valid for the current selection mode. If the selection mode is CONTIGUOUS\_TREE\_SELECTION and a RowMapper exists, this will make sure all the rows are contiguous, that is, when sorted all the rows are in order with no gaps. If the selection isn't contiguous, the selection is reset to contain the first set, when sorted, of contiguous rows.

If the selection mode is SINGLE\_TREE\_SELECTION and more than one TreePath is selected, the selection is reset to contain the first path currently selected.

### arePathsContiguous

protected boolean **arePathsContiguous**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] paths)

Returns true if the paths are contiguous, or this object has no RowMapper.

### canPathsBeAdded

protected boolean **canPathsBeAdded**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] paths)

Used to test if a particular set of TreePaths can be added. This will return true if paths is null (or empty), or this object has no RowMapper, or nothing is currently selected, or the selection mode is DISCONTIGUOUS\_TREE\_SELECTION, or adding the paths to the current selection still results in a contiguous set of TreePaths.

### canPathsBeRemoved

protected boolean **canPathsBeRemoved**([TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)[] paths)

Returns true if the paths can be removed without breaking the continuity of the model. This is rather expensive.

### notifyPathChange

protected void **notifyPathChange**([Vector](http://docs.google.com/java/util/Vector.html)<javax.swing.tree.PathPlaceHolder> changedPaths,  
 [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) oldLeadSelection)

Notifies listeners of a change in path. changePaths should contain instances of PathPlaceHolder.

### updateLeadIndex

protected void **updateLeadIndex**()

Updates the leadIndex instance variable.

### insureUniqueness

protected void **insureUniqueness**()

This method is obsolete and its implementation is now a noop. It's still called by setSelectionPaths and addSelectionPaths, but only for backwards compatability.

### toString

public [String](http://docs.google.com/java/lang/String.html) **toString**()

Returns a string that displays and identifies this object's properties.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a String representation of this object

### clone

public [Object](http://docs.google.com/java/lang/Object.html) **clone**()  
 throws [CloneNotSupportedException](http://docs.google.com/java/lang/CloneNotSupportedException.html)

Returns a clone of this object with the same selection. This method does not duplicate selection listeners and property listeners.

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a clone of this instance. **Throws:** [CloneNotSupportedException](http://docs.google.com/java/lang/CloneNotSupportedException.html) - never thrown by instances of this class**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultTreeSelectionModel.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/tree/DefaultTreeModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/tree/ExpandVetoException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/tree/DefaultTreeSelectionModel.html)    [**NO FRAMES**](http://docs.google.com/DefaultTreeSelectionModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#4d34og8) | [CONSTR](#3j2qqm3) | [METHOD](#4i7ojhp) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).